

To: David S. Bayard (Fiscal Year 2004 GN&C Small Body
Proximity Operations & Landing R&TD)
From: Adnan Ansar
Subject: Small Body GN&C Research Report: Feature Recognition Algorithms

ABSTRACT

We present techniques for feature recognition on small bodies using camera imagery, specifically with the aim of matching acquired features against a pre-existing catalog of general landmarks. This recognition capability is a necessary input to the navigation filter and is also of use in reacquiring areas of scientific interest for further examination or for sample return. We describe the feature detection mechanism and the descriptors used to identify features as well as various techniques to enhance invariance properties under imaging conditions likely to be encountered during a small body mission. We also describe additional computer vision products of use to a small body mission which fit within the feature detection framework.

1 Introduction

The image processing work in support of this R&TD task ultimately centers on generating the landmark table (LMT) and auxiliary landmark table (LMTX) for each image of the target body acquired by the spacecraft using **general landmarks**. We describe in detail (1) the identification of candidate landmarks, (2) their image localization, which corresponds to the bearing angles z_α and z_β in LMT, and (3) the feature descriptors d_I recorded in LMTX and used to match candidate features against their descriptors d_C in the feature catalog (FCAT). A superset of those features identified as landmarks can be used in the paired feature table (PFT) and auxiliary paired feature table (PFTX) for frame to frame motion estimation. Alternatively, simple feature tracking independent of the general landmarks may be used. The latter is a thoroughly understood technique which, depending on extent of motion, can be performed at video framerate (30Hz) or faster. In the following, we focus exclusively on the general landmarks used to populate LMT and LMTX.

Past work on landmark identification and matching has focused on geometric primitives in image data, which can be reliably parameterized by a simple model. In particular, the Machine Vision Group has done extensive work on identification and matching of craters[1], which appear as ellipses in imagery. Information on sun angle and shading phenomena is used to make the resulting product largely insensitive to lighting conditions. While this crater detection work has been very successful, it's applicability is limited to bodies with well-defined (i.e. uneroded) craters. In cases where craters are more jagged or where no craters are present, it cannot be used. Since the likely target of future small body missions will include such objects, some attempt must be made to address this deficiency. Possibilities include widening the set of geometric primitives or using differential (curvature, locations of cusps, etc) and topological information on curves and curve intersection.

However, it is still possible that an unexplored small body will fit none of the requirements for a specific model-based landmark. It then becomes essential to develop a type of **general landmark** that is model-free and depends only on local image information. The algorithm we describe below is based on the notion of selecting features automatically at some optimal scale and is adapted from David Lowe's[2] scale invariant feature tracking (SIFT) algorithm. In a broad sense, this is a generalization of Harris corner features[3]. We find points of interest in scale space[4], a stack of band-pass filtered copies of an image, and record oriented gradient information in the scale neighborhood of a selected feature. The interest point becomes a candidate landmark, and the gradient information becomes the basis for a recognizable descriptor.

The descriptor, d_I in LMTX, is compared to FCAT. If a matching d_C is found, then the landmark is considered identified. If not, the new landmark may be used to update FCAT. In general, FCAT should satisfy the following requirements.

1. It must be dynamically updated to admit newly identified landmarks.
2. It must be efficiently organized for easy search.
3. It must be insensitive to changes in viewpoint, scale and illumination.

The update mechanism and efficient organization of the catalog will be topics of future work, and we assume in the following that a catalog has already been populated. Much of our focus for

the current year has been on algorithm development addressing the item 3 above. Model-based approaches are inherently viewpoint and scale invariant, subject to the geometry of the primitives used. In the case of craters, the projective invariants of conic sections are well understood, and the explicit modeling of lighting angle affords a degree of illumination invariance. In the case of general landmarks, scale invariance is built into the descriptor selection mechanism, as is a certain amount of viewpoint invariance. We will describe in detail efforts to address viewpoint and illumination invariance.

Using image data, we are able to supply information beyond identification of landmarks in the catalog. Given knowledge of 3D locations of landmarks, we can also extract from a single frame (subject to image and map noise) the 6 DOF position and attitude of the camera with respect to the body. While this information may be of use as a sanity check for the state estimator, it is immediately relevant from a vision standpoint as a means of using geometric rigidity to remove false matches between LMT and FCAT. If 3D landmark locations are unknown, we can use two or more frames to recover 3D structure and relative camera motion between frames up to an unknown scale. Extensive work has been done in computer vision in these areas.

We now describe general landmarks in some detail.

2 General Landmarks

General landmarks are image features that can be identified across changes in viewpoint, distance to object and illumination conditions. In our case, they are not model based but are derived purely from image data. The specific type of landmarks we consider are based on David Lowe's SIFT features. Since the natural setting for these features is scale space, we begin with a brief overview of the topic.

2.1 Scale Space

We omit much of the underlying theory of scale space and describe only some relevant aspects. Details can be found in [4, 5]. Our goal is to find image features reliably across a variety of image scales, corresponding to different distances between the camera and object. Associated with any feature in a given image is some inherent scale. In order to compare to the same feature in another image with a different inherent scale, it is necessary to place both in the proper framework. Thus, if $I(x, y)$ is the gray value of an image at point (x, y) , we consider

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) = \int \int G(x - \xi, y - \zeta, \sigma) I(x, y) d\xi d\zeta$$

the convolution of the image with a Gaussian kernel $G(x, y, \sigma)$ given by

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Given the same feature in images I_1 and I_2 at locations $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, respectively, we expect that there exist σ_1 and σ_2 such that local image properties of $L_1(x_1, y_1, \sigma_1)$

are similar to those of $L_2(x_2, y_2, \sigma_2)$ in appropriately sized neighborhoods centered on p_1 and p_2 . Intuitively, this means that modulo orientation and rescaling, the same feature in two images differs only by a Gaussian blur dependent on relative scale.

While $L(x, y, \sigma)$ is used to construct the descriptor for a feature, its image location is actually identified in a different space. Let

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

for some fixed choice of k . This is clearly equivalent to convolution of $I(x, y)$ by a Difference of Gaussians (DoG). Features correspond to extrema in this 3 dimensional space $D(x, y, \sigma)$. There are several ways to interpret this choice. The collection of $D(x, y, \sigma)$ for all σ can be considered a collection of bandpass filtered copies of the image, where k controls the size of the pass band. If a feature (defined as an extremal value in D at some spatial frequency) is identified in one image, the corresponding point viewed at a different scale in a different image should also exhibit an extremum at its optimal spatial frequency. An explicit search in D will find all points which exhibit this behavior.

Alternatively, it can be shown that the DoG approximates the scale normalized Laplacian of Gaussian (nLoG) $\sigma^2 \nabla^2 G$ as $k \rightarrow 1$. This terms approaches the derivative of G with respect to σ . The explicit connection between $\frac{\partial G}{\partial \sigma}$ to the nLoG is made via a partial differential equation similar to the heat diffusion equation, but parametrized by σ rather than time. Details can be found in [2]. The extrema of the nLoG has been shown empirically to be a stable class of scale invariant image feature[6] under various geometric transformation. The approximation D to the nLoG as used by Lowe has the advantage of simplifying some of the later computations.

2.2 Summary of algorithm

As mentioned above, our features correspond to extrema of the discretized DoG space $D(x, y, \sigma)$ with separation defined by k . Once a feature is identified, we sample gradient data on an oriented patch in the corresponding slice of $L(x, y, \sigma)$ (i.e. with the same σ as the detected feature) of scale space centered on the feature. This forms the basis for the feature's scale invariant descriptor.

2.2.1 Details of Feature Detection

We base our implementation on the algorithm overview in Lowe's paper[2] and refer the reader to that paper for more detail. We also describe our own task specific modifications to the algorithm.

Images scales $L(x, y, \sigma)$ are computed for

$$\sigma = k^n \sigma_0$$

where σ_0 represents an initial smoothing of the image, k is the fixed constant described above, and n ranges over integers from 0 to some maximum. For computational efficiency, both in creation of the scales and in eventual determination of the feature descriptors, the scale space is organized into octaves. Within each octave are a fixed number, say f , of steps with the combination of f and k chosen so that $k = 2^{1/f}$. This implies that from one octave to the next, the change in scale is

$$k^f = (2^{1/f})^f = 2$$

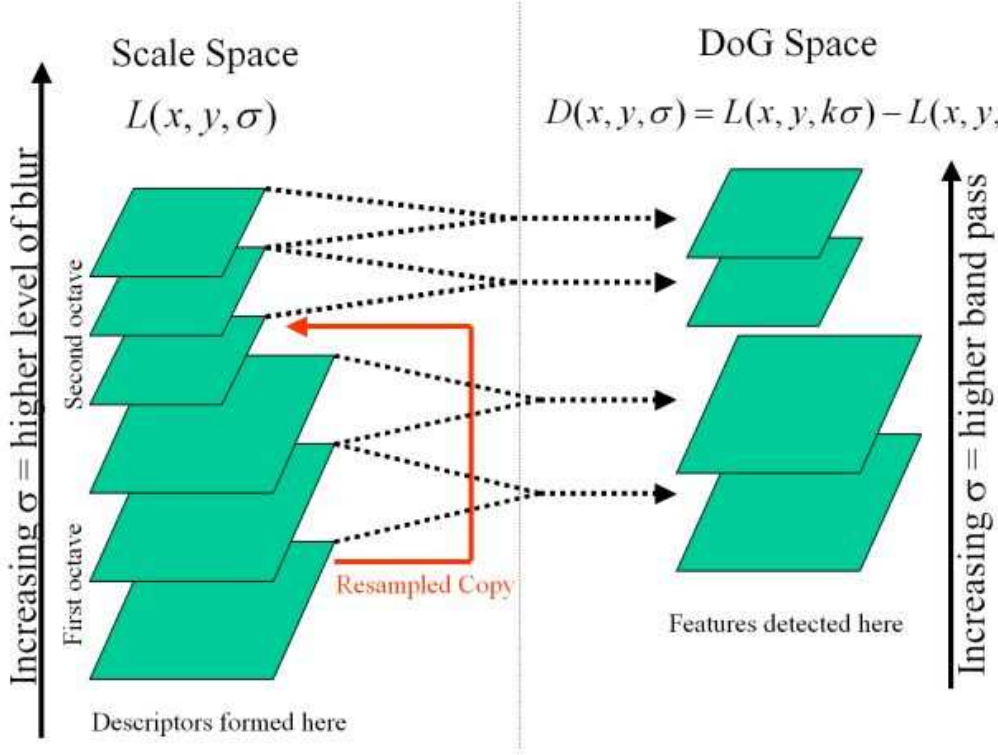


Figure 1: Diagrammatic representation of scale space L and DoG space D . Successive octaves are formed in L by resampling. Features are detected in D and their descriptors are formed from the corresponding scale slice in L .

However, a scale change from σ to 2σ can be approximated by a factor of 2 smooth resampling of $L(x, y, \sigma)$. Thus, $L(x, y, 2\sigma)$ is constructed with spatial dimensions half the size of $L(x, y, \sigma)$. This has the advantage of (1) requiring explicit computation of Gaussian convolutions for only the first scale, and (2) reducing the size of the scale space, since each octave requires a fourth the storage of the preceding octave. This pyramidal structure for the scale space has a natural simplifying implication for the construction of the feature descriptor. Since succeeding octaves are smaller, the feature patch used to compute a descriptor can be kept a fixed size throughout while still covering relevant image data. In effect, the descriptor window (always a fixed size) of a feature viewed from close will have the same information content as the window of a feature viewed from far, since the close object will be resized to something approaching the distant object for some appropriate scale.

Once the scale space is computed, we compute $D(x, y, \sigma)$ by differencing adjacent steps of the scale space. These ideas are illustrated in graphical form in Fig. 1, which is adapted from [2]. The only problem occurs at boundaries of octaves, where the sizes of the first step of a given octave and the last step of the preceding octave do not match. We solve this simply by resampling the larger of the two steps. We show an example of both $L(x, y, \sigma)$ and $D(x, y, \sigma)$ for an image of the

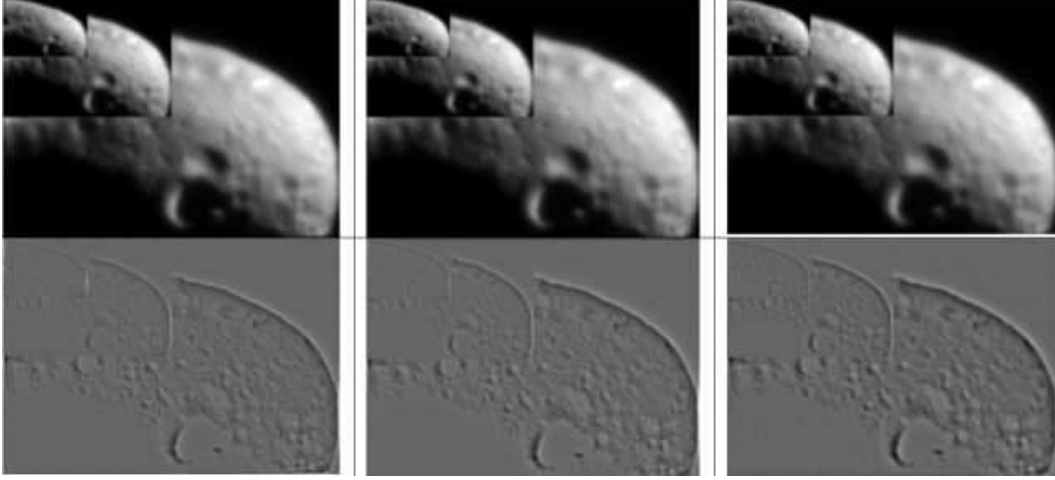


Figure 2: Visualization of scale space (top row) and DoG space (bottom row) for an image of the Eros asteroid. The largest images across each row represent the first octave, the next largest the second, and the smallest the third. In any given octave, each step from left to right adds a factor of k Gaussian blur in the scale space or a factor of k shift in pass band for the DoG space.

asteroid Eros in Fig. 2. We use only 3 steps in each of 3 octaves for easier visualization. In our construction above, there are 3 parameters that define the scale space and DoG space. These are the initial smoothing (σ_0), the number of octaves, and the number of steps per octave (f). Observe that the choice of f defines the scale separation by $k = 2^{1/f}$.

Having computed the discrete difference of Gaussian space $D(x, y, \sigma)$, we perform an explicit search for extrema by examining for each coordinate (x, y, σ) its 26 neighbors. Again, there is a slight complication at octave boundaries because of mismatch in image size. This is solved either by resampling the image up or down by a factor of 2, as the circumstance dictates. Once a preliminary list of extrema is found, a series of tests must be performed to eliminate those that are poor candidates for feature matching.

Suppose an extremum of the DoG function D is found at (a, b, γ) . The following tests are employed:

- We threshold on the absolute value of D at (a, b, γ) . This tends to eliminate the largest number of unstable features. Our current implementation accepts only values above some user-defined fraction of the median of values of D at all extrema.
- We also suppress features in areas of low image variance. This is done by computing pixelwise image statistics at each scale and masking out regions of low variance. We must, therefore, examine the image statistics at $D(a, b, \gamma)$ before accepting a point as a valid feature.
- Since the LoG, hence DoG, has a strong response to all edges, we need to eliminate “edge-like” features in favor of “corner-like” features, since the former is poorly localized. This

is in rough analogy to the Harris corner detector. The eigenvalues of the 2D Hessian matrix $H(x, y)|_{(x,y)=(a,b)}$ at (x, y, γ) are the principle curvatures of the surface defined by $D(x, y, \gamma)$ at (a, b, γ) . If one principal curvatures differ greatly in magnitude from the other, then the candidate point likely belongs to an edge and is a poor choice for a feature. A trick employed by Lowe and borrowed from Harris is to compare the ratio of $\text{tr}(H)^2$, the square of the trace, to $\det(H)$, the determinant. If the eigenvalues of H are e_1 and e_2 with $e_1 = re_2$, then

$$\frac{\text{tr}(H)^2}{\det H} = \frac{(e_1 + e_2)^2}{e_1 e_2} = \frac{(re_2 + e_2)^2}{re_2^2} = \frac{(r + 1)^2}{r^2}$$

The left hand side achieves a minimum when $r = 1$ and increases monotonically with r . Hence, there is no need to compute eigenvalues explicitly. We need only threshold on the ratio on the left.

Finally, we fit each extremal point (a, b, γ) of D to a quadratic Q in (x, y, σ) such that.

$$Q = \min_{\tilde{Q} \in P_2(x,y,\sigma)} \sum_{x=a-1}^{a+1} \sum_{y=b-1}^{b+1} \sum_{\sigma=\frac{1}{k}\gamma}^{k\gamma} |\tilde{Q}(x, y, \sigma) - D(a, b, \gamma)|$$

where $P_2(x, y, \sigma)$ is the space of 2^{nd} degree polynomials in (x, y, σ) . If Q achieves an extremum at (x_o, y_o, σ_o) , we apply the following criteria.

- If (x_o, y_o) differs by more than one pixel from (a, b) or γ differs by more than one step from σ_o , the candidate is rejected.
- If (x_o, y_o) differs by less than one pixel but more than one-half pixel from (a, b) or γ differs by less than one step but more than one-half step from σ_o , we move the candidate in the appropriate spatial and scale directions and refit the polynomial.
- If (x_o, y_o) is within one-half pixel of (a, b) and σ_o within one-half scale of γ , we accept (x_o, y_o, σ_o) as the subpixel location of the feature. Finally, given the camera model, these pixel coordinates can be transformed into 3D coordinates on the CCD of the camera. From this, the bearing angles z_α and z_β reported in LMT (or PFT) can be obtained. σ_α and σ_β in LMT will be derived from combination of known sensor noise and the quality of the subpixel fit (i.e. the residual in fitting Q to the data.)

2.2.2 Feature Descriptor

Once a feature is detected in the DoG space, we must develop a descriptor for it that can be recognized under changes in viewing condition and scale. The descriptor appears in LMTX and PFTX as d_i for frame i . A description of the procedure follows.

The feature descriptor is based on local gradient information at optimal scale. Since we seek a viewpoint independent solution, the first step is to eliminate dependence on image orientation. This is accomplished by computing a principal orientation around each feature point in the DoG space

and reorienting the feature according to this information. If the feature is found at (a, b, γ) , we poll the 2 dimensional gradient of $D(x, y, \gamma)$ near (a, b) . Let $M(x, y)$ be the magnitude and $O(x, y)$ the orientation of $\nabla_{x,y} D$. We construct an orientation histogram of N bins, each occupying $360/N$ degrees in the image plane as follows. If $O(x, y)$ is q degrees, then we round q/N and add to the corresponding bin in the orientation histogram the value

$$M(x, y) \exp\left(-\frac{(x-a)^2 + (y-b)^2}{\lambda^2}\right)$$

for some λ . This is the magnitude of the gradient at (x, y) attenuated by a Gaussian centered at (a, b) . The choice of λ determines the radius around (a, b) used to poll for the histogram. We typically use a radius of $3 * \lambda$ and set $N = 36, \lambda = 3$. The mode of the histogram then indicates the local orientation of the feature. If the histogram is bimodal with a second mode at least 80% as large as the primary mode, then we record both orientations and construct two features at the same coordinate, one with each local orientation.

Given the orientation above, we rotate the region around the feature using bilinear interpolation to a canonical orientation of 0. The result is complete rotation invariance in the image plane, up to the orientation resolution N . The size of the neighborhood rotated is dependent on the size of vector we use for the feature descriptor. We now describe this.

Since all features orientation can be rotated to 0, it suffices to construct a descriptor based on a grid aligned with the (locally re-oriented) image axes. Thus, assume without loss of generality that the orientation of a feature at (a, b, γ) in the DoG space is 0. We partition $D(x, y, \gamma)$ in a neighborhood of (a, b) with radius R into a $K \times K$ grid. In each grid cell, we compute a histogram of local orientations using L bins per cell. This is done essentially as above for the principal feature orientation, except that we do not attenuate the gradient magnitudes by a Gaussian. The $L \times K \times K$ list of histogram values taken in some predefined order is then the feature descriptor. Typical values used are $R = 8, L = 8, K = 4$, for a 128 dimensional vector representing each feature, using information from a 16 pixel x 16 pixel region centered at (a, b) in $D(a, b, \gamma)$. Data a distance greater than R from (a, b) is ignored, producing a circular mask on the descriptor. See Fig. 3 for a graphical representation, borrowed from Lowe, of this descriptor using the typical parameters indicated above. This scheme affords a certain degree of shift invariance. Observe that if the location of the feature shifts by a small amount in any direction, there is relatively little change in the histogram of any cell in the grid, since the majority of pixels contained in it remain unaffected. Thus, small shifts due to viewpoint change can be easily accommodated.

Once the descriptor is computed and entered into LMTX, we compare to the list of feature descriptors in FCAT. At the moment, we use a simple threshold on Euclidean distance between 128 dimensional vectors as well as some geometric constraints described below. We will eventually use more sophisticated data structures to organize FCAT for easy search. The PCA version of the algorithm (described below) places an inherent organizational structure on the catalog, as we will see.

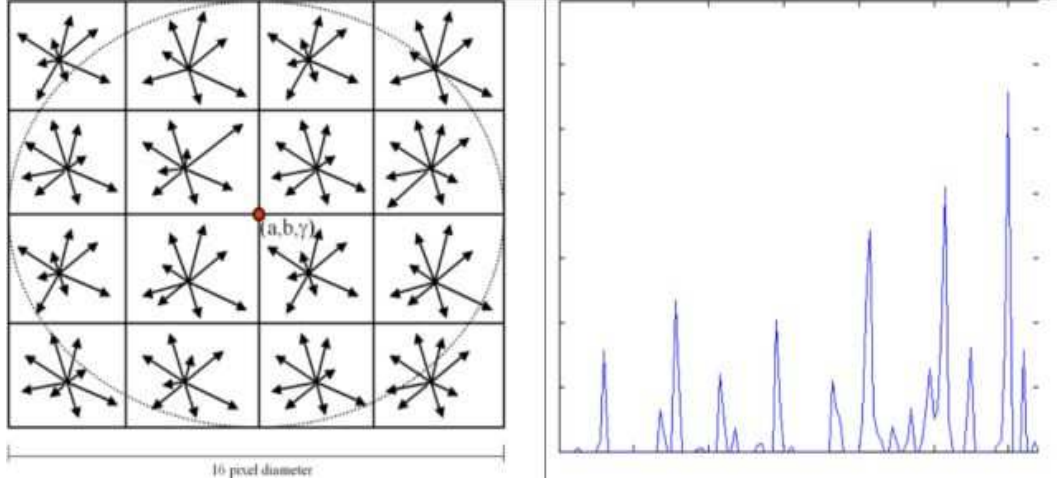


Figure 3: On the left is a graphical representation of feature vector. The length of each arrow corresponds to the weight of a given orientation in each of the 4x4 grid cells. These concatenated lengths represent the feature descriptor. On the right is an actual descriptor in graphical form from a feature on the Eros image in Fig. 2

2.2.3 Invariance Properties and Limitations

By construction, the descriptors are invariant to Euclidean motions of the image plane, at least up to discretization. In other words, any combination of rotation and translation of the image plane is acceptable. The use of the scale space formalism adds scale invariance. Thus, any transformation of the form

$$\begin{pmatrix} \cos(\theta) & \sin(\theta) & t_x \\ -\sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

applied to homogeneous image coordinates $(x, y, 1)^T$ can be accommodated. For the small body task at hand, there is a potentially wider range of image transformations. For successive frames during orbit, the above is often a good approximation. However, as viewpoints change dramatically (possibly over multiple orbits), the transformation of the image plane is poorly approximated by the scaled 2D rigid motion above. In Fig. 4, we show matching results for adjacent frames and for frames differing by one full orbit of the NEAR spacecraft around Eros. In both cases, there is relatively little out of plane rotation. Thus, features are successfully matched. In Fig. 5, we illustrate scale invariance using imagery of the asteroid Mathilde taken by the NEAR spacecraft during approach.

Another key issue is illumination invariance. In the examples shown in Fig. 4, illumination conditions have changed somewhat in the image pair on the right. However, much larger variation can be expected in general. In particular, imagery of objects in space tends to exhibit both high contrast and dramatic contrast change as a function of change in illumination direction. The algorithm as it stands is largely incapable of handling such variations in illumination. Even the imagery

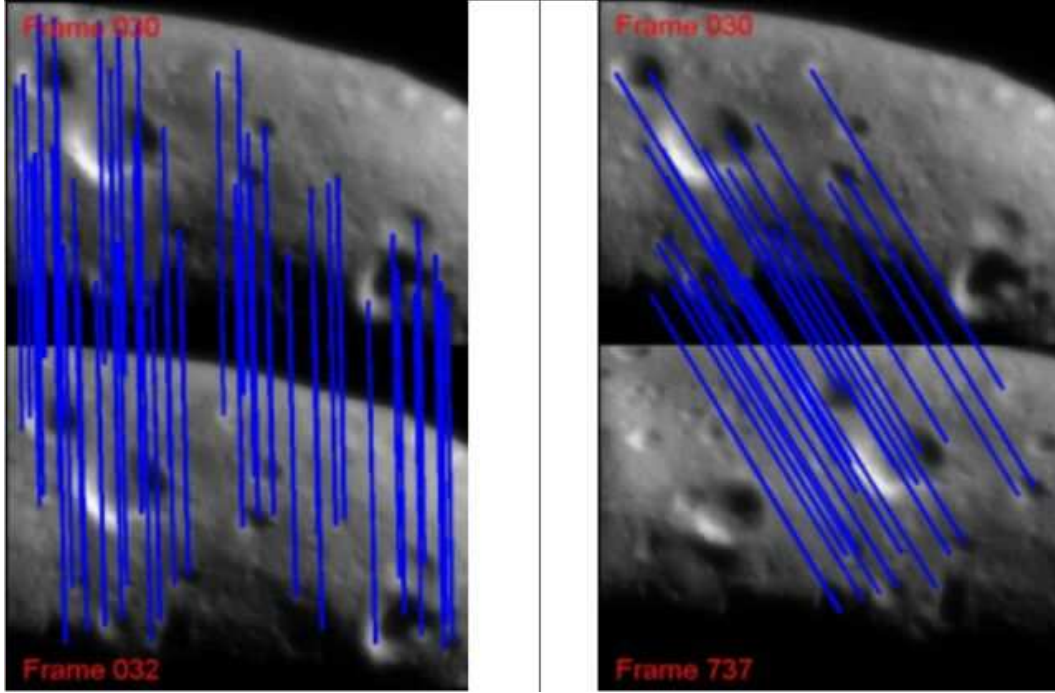


Figure 4: Feature matches across adjacent frames and after one full orbit of NEAR spacecraft around Eros.

in Fig. 5 presented problems for the base algorithm.

One solution to both the viewpoint and illumination problems is to develop FCAT so that it includes landmarks from the same area on the target viewed at multiple illumination conditions and from several directions and save all descriptors. Thus the descriptors in LMTX will match landmarks from at least some iteration of the process. This may be possible in some mission scenarios and impractical in other. For the latter case, we must still try to improve viewpoint and illumination invariance for features. We describe our attempts thus far below.

2.3 Addressing Viewpoint and Illumination Invariance

We describe techniques to address the problems of out of plane rotation and illumination change. In a general setting, only image based information would be known, and we explore this first. However, if some state information is also known, more can potentially be done.

2.3.1 Image based solutions

We describe techniques to address the problems of out of plane rotation and illumination change. In a general setting, only image based information would be known, and we explore this first. However, if some state information is also known, more can potentially be done.

The base algorithm is not designed to handle large changes in viewpoint given objects with highly 3 dimensional (i.e. nonplanar) structure in the scene. So we focus only on illumination in-

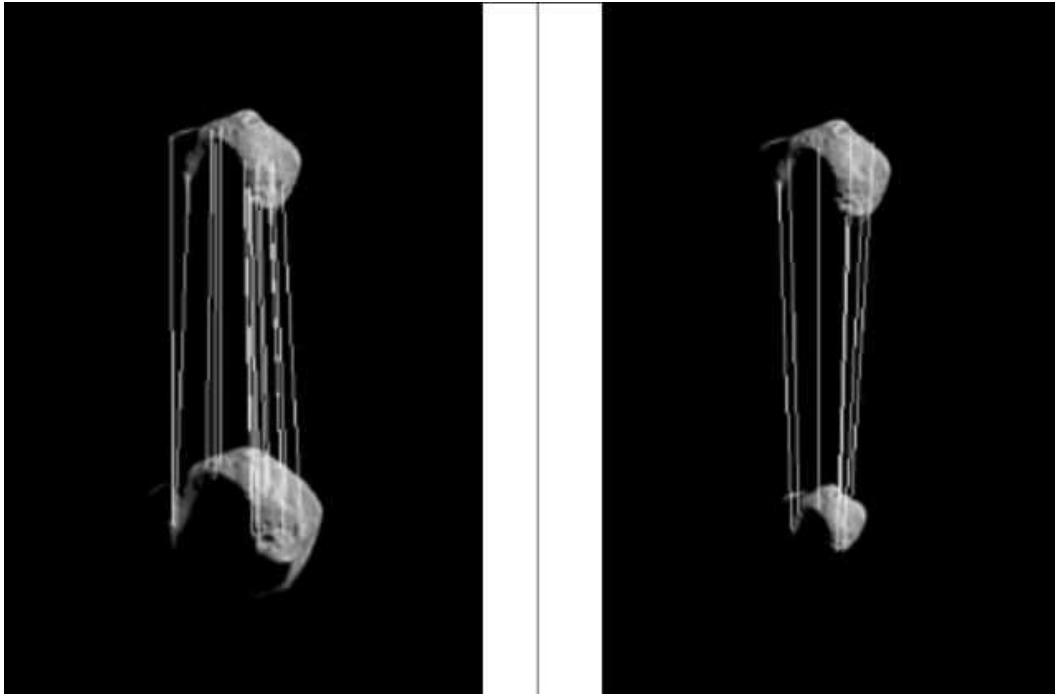


Figure 5: Scale invariance is demonstrated by imagery of the Mathilde by the NEAR spacecraft. The same image is used on the top half of each frame. The bottom half is an image of Mathilde at roughly double (left) and half (right) this size.

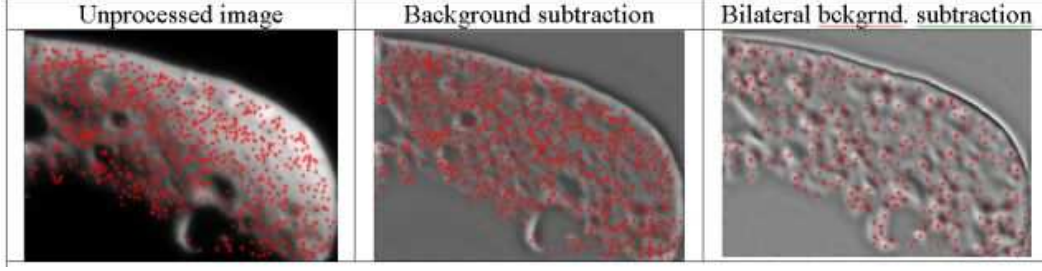


Figure 6: . Features for an image of Eros without preprocessing, with block average background subtraction, and with bilateral background subtraction. The intensity normalization of the two preprocessing techniques often results in greater illumination invariance.

variance using image based solutions. To a certain extent, the use of the DoG space with inherent frequency bandpasses limits the dependence of the data on lighting. Lowe makes a further explicit attempt to eliminate illumination dependence by truncating the orientation histograms used to build the feature descriptor. A maximum is assigned for all bins, and any bin that exceeds the maximum is set to the maximum value. This is helpful in cases where saturation leads to very high gradient values. However, it has little effect on more subtle but equally destructive variations in illumination, particularly with the high contrast imagery we are using. We have had some success with preprocessing the image by background subtraction. This amounts to subtracting from each pixel the average gray value of a large (in our case 31×31) block centered on the pixel. If $I(x, y)$ is the image, we replace with $J(x, y)$ given by

$$J(x, y) = I(x, y) - I(x, y) * F(x, y)$$

where F is an $n \times n$ matrix with $F(i, j) = \frac{1}{n^2}$. In some cases, we have had greater success with bilateral background subtraction, in which a bilateral filter is used in place of a block average. If $Bf(I)$ is the filtered image, we use

$$J(x, y) = I(x, y) - Bf(I)(x, y)$$

to compute the scale space. The later (see [7] for details) is an edge preserving smoother which does a better job of maintaining useful texture information near crater rims and other sharply defined structures. We omit details and refer the reader to the paper cited above. The frames in Fig. 5 were processed using bilateral background subtraction, since the raw imagery failed to produce many matches. Either technique effectively precedes the construction of the scale space and DoG with a highpass filter of the image. In Fig. 6, we show the results of both block average and bilateral background subtraction on an image of Eros, as well as the features detected with each subtraction method and on the raw image. Observe in this case that while the bilateral result reduces the overall number of features, they are typically more salient than those detected otherwise.

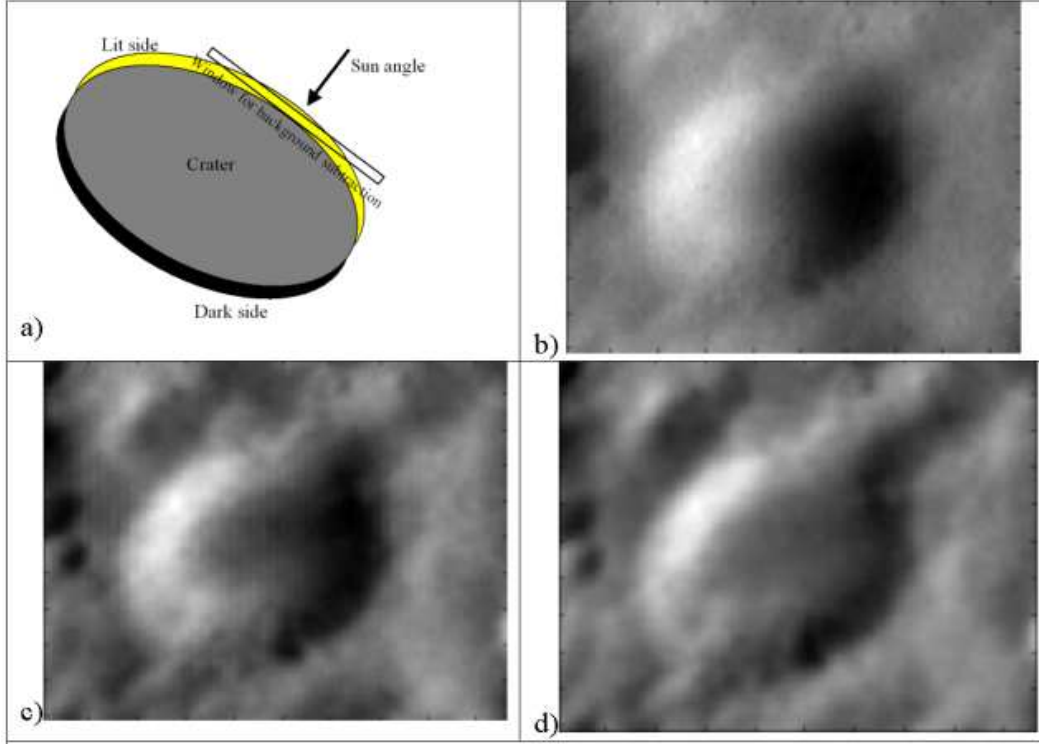


Figure 7: . a) Diagram showing use of sun angle for background subtraction. b) Image of real crater from Eros. c) Result of bilateral background subtraction. d) Result of bilateral background subtraction using sun angle and thin rectangular window. Observe that more of the crater interior is usable in frame d)

2.3.2 Using State Information

FCAT is intended to contain information on sun angle (s), approximate viewing direction (v) and approximate viewing distance (v_d). A simple but useful piece of information is the sun angle in the image. If this is known, we can try to compensate for shadow effects in illumination direction. Rather than the usual block average or bilateral background subtraction, we use a thin filter window oriented perpendicular to the sun direction. A simple technique is to rotate the image until the sun angle is vertical and then to perform background subtraction using a thin rectangle aligned horizontally. This averages over regions of highlight or darkness due to the interplay between 3D structure and lighting, while minimizing any averaging over regions of sharp lighting variation. We illustrate the idea in Fig. 7 and show results for bilateral background subtraction both with and without knowledge of sun angle. Invariance to illumination is one of two major challenges. The other is invariance to viewpoint change. While the base algorithm is quite insensitive to rotations and shifts in the image plane, perspective distortion resulting from camera motion in 3D is much harder to accommodate. We have attempted to introduce a degree of affine invariance as follows. If we know approximately the center of gravity G of a roughly spherical body, the viewing direction V of the camera, and the position C of the camera in some inertial reference frame, we can attempt to compensate for foreshortening effects. This is accomplished by compressing the image in a

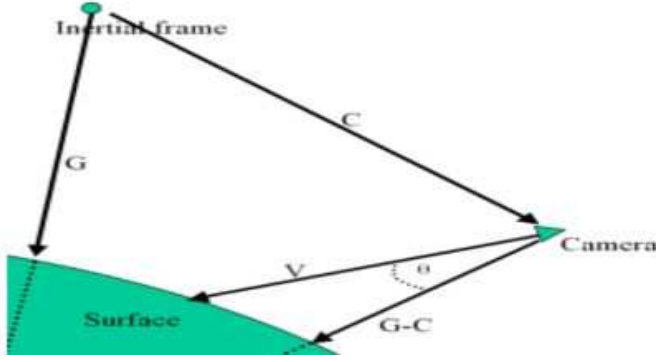


Figure 8: Proposed technique for reducing foreshortening effects prior to calculating feature descriptor. The image is stretched by a factor of $\cos(\theta)$ in the direction corresponding to the projection of $G-C$ onto the image plane

direction corresponding to the projection of $G - C$, the vector from the camera to the center of gravity of the body, onto the image plane by an amount depending on the cosine of the angle between the $G - C$ and V . This is illustrated in Fig. 8 by a 2D sketch. The approach becomes a more realistic approximation to the true perspective distortion only as the curvature of the surface approaches zero, in other words when the surface can be locally approximated by a plane and $G - C$ approaches the surface normal. This assumption may be applicable even to small bodies during close orbit or descent. However, our tests with imagery of Eros were unsuccessful. We suspect that the irregularity of the surface and the relatively high orbits used in our test imagery played a factor. This topic is explored further in Sect. 2.3.3.

Finally, we use multi-frame cues to increase the likelihood of matches between images taken at different viewpoints. If two features can be connected through a chain of images, they can be matched even if their individual (i.e. single frame based) descriptors fail to match directly. This is a simple case of a planned dynamic update scheme for the FCAT, in which the descriptor of a feature is modified in some weighted average fashion as it is reacquired in successive frames.

We show the result of using both sun angle information for oriented background subtraction and multi-frame cues in Fig. ???. Observe that while single frame-to-frame matches begin to fail quickly, the use of sun angle and multiple frames allows matches across much greater changes in viewpoint and lighting. This is a good indication that a dynamic update of descriptors in the feature catalog will increase match likelihood. Note that while we only display matches for every fourth frame in Fig. ??, we use information from intervening frames in the computation.

2.3.3 3D reconstruction and Epipolar constraint

Surface irregularity is relevant for two reasons. First, we can use it to judge the applicability of the foreshortening compensation described above. More fundamentally, we can use it as a measure of the quality of a landmark. This is applicable not only to constructing LMT and LMTX but to populating FCAT. A feature in an area of high 3D structure is unlikely to be a stable landmark

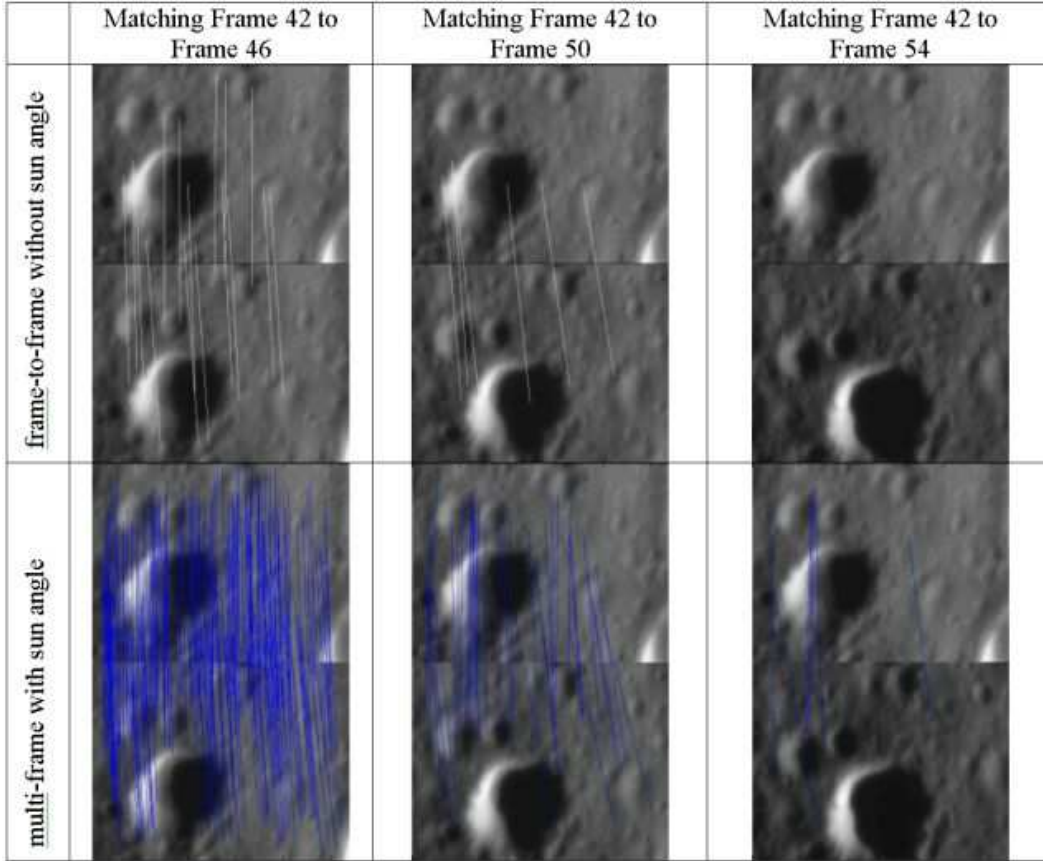


Figure 9: We show matches for a short sequence of Eros images in which viewpoint and lighting conditions change over time. In the top row, the base algorithm is used without information on sun angle and without multi-frame cues. In the bottom row, we use both the sun angle and multi-frame cues. Observe that in the last frame (right), the base algorithm fails entirely while the modified algorithm matches several features.

under either viewpoint or illumination change. Fortunately, we are able to compute 3D structure up to scale from two adjacent views. Use the matches found in PFT to start the process.

Given two adjacent or nearby frames and a series of feature matches supplied by the PFT, we show how to reconstruct the surface and determine whether a region is a good candidate for landmark placement. As part of the process, we will describe in brief the epipolar camera geometry and its implications for matching features either in the PFT or between FCAT and LMT.

Given a set of point correspondences (such as that supplied by PFT) between two frames, there is a constraint on the 3D location of the imaged points and their image coordinates. For any point P in 3D, consider the plane Pl formed by that point and the projection center of the camera in the two positions at which the two frames were taken, say at times t_1 and t_2 . It can be shown that the imaged point in either frame lies on the intersection of Pl with the CCD at the time of image capture. Let p_1 and p_2 be the 3D coordinates of the projected point on the CCD at times t_1 and t_2 in the coordinate frame of the camera. We can compute these points from image coordinates using the camera model. Borrowing terminology from projective geometry, this is known as the epipolar constraint. Suppose P is expressed in the camera frame at time t_1 . In the absence of noise, there exists a 3×3 matrix E , called the essential matrix, such that

$$p_1^T E p_2 = 0$$

E is given by $E = RT_{[\times]}$. This constraint captures the epipolar geometry in its entirety. Here (R, T) is the Euclidean motion undergone by the camera frame between time t_1 and t_2 , and $T_{[\times]}$ is the skew symmetric matrix associated with cross product by T (i.e. for a vector v , $T_{[\times]}v = T \times v$). Given a set of point correspondences, the relation above yields a (generally overdetermined) linear system in the entries of E , which we can solve by various means such as singular value decomposition. Once E is known, it can be decomposed into R and $\bar{T} = \kappa T$, where \bar{T} has unit norm. Without an additional constraint on scale, κ cannot be determined. This is clear from the epipolar constraint, since any rescaling of T , hence E , has no effect. We omit the details of this procedure and refer the reader to [8]. Once the linear estimate of the motion is obtained, we typically refine it with a non-linear optimization that minimizes the reprojection error to simultaneously solve for motion and 3D structure.

The above computation not only gives us 3D reconstruction, it provides a quick way to check for false matches between the descriptors in LMTX and FCAT. False matches arise from similar descriptors for two features. However, the 3D location of those features and their projections are generally inconsistent with the epipolar geometry. Thus, we compute E using subsets of the full dataset and a robust technique such as RANSAC [9]. We then check the value of $p_1^T E p_2$ for each candidate match in either the PFT or between the LMT and FCAT to identify outliers.

Once the relative motion is known, we can reconstruct the scene geometry using stereo vision techniques (see [8] for details). With this data, we can do local plane fitting to determine surface normals and correct foreshortening. We can also get a good sense of the roughness of a region independent of absolute scale by computing the local variance in stereo range values. In Fig. 10 we show a synthetic, texture mapped scene, the resulting range map from applying the procedure outlined above, and a roughness map computed as log of the range variance in a 3×3 window centered on each pixel. Note that areas with high 3D relief are clearly visible and can be avoided for both entry into the LMT and populating the FCAT.

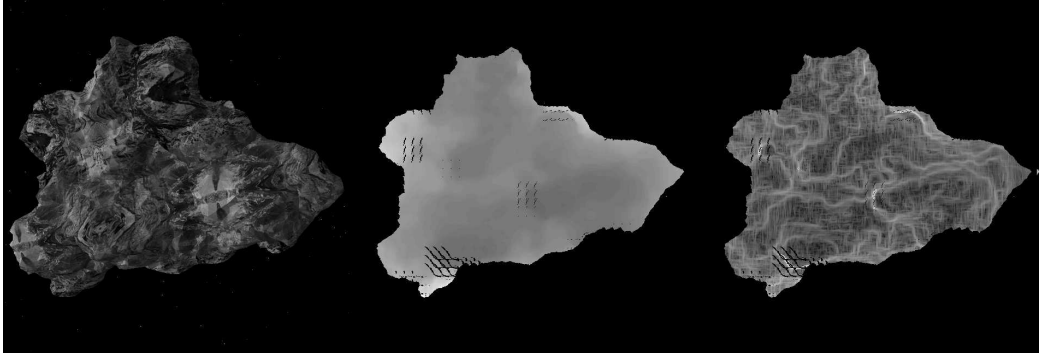


Figure 10: A synthetic scene, the range map computed from PFT and techniques described in Sect. 2.3.3, and a roughness map computed from the range. The roughness can be used to guide the selection of good landmarks for viewpoint and illumination invariance.

2.4 Principal Component Analysis

Principal Component Analysis (PCA) is a technique for extracting the most relevant subspace for data distributed in a high dimensional vector space, such as our feature descriptors. In brief, we compute the eigendecomposition of the cross covariance matrix of the dataset and project the data onto a basis consisting of the eigenvectors. Then the most relevant directions correspond to the largest eigenvalues, and we can truncate the vector representation of the data in the new basis without losing important information. Given an n dimensional dataset and the normalized eigenvectors, say e_1, e_2, \dots, e_m , corresponding to the largest m eigenvalues, we can project any vector $v \in R^n$ into its most significant m components by $v' = Pv$, where $v' \in R^m$ and

$$P = \begin{pmatrix} \dots & e_1 & \dots \\ \dots & e_2 & \dots \\ \vdots & \vdots & \vdots \\ \dots & e_m & \dots \end{pmatrix}$$

Following some recent work in the literature [10], we are testing a PCA version of the feature algorithm. Unlike the algorithm described above, the PCA algorithm has two phases. The first is a computationally intensive training phase needed to determine the projection matrix P . We expect that this can be done offline on data collected by the spacecraft and sent to Earth. The second is actual feature detection and matching. We first describe the training phase

We first collect information on a large set of features, possibly over several orbits of the body. The base algorithm proceeds as described above up to determination of a principle orientation for each feature. We then rotate a large $k \times k$ neighborhood of the detected feature in the correct scale slice of the DoG space to set the principle orientation to zero, again as above. Typically, the neighborhood is greater in size than 30×30 . Now we simply poll the gradient orientations and magnitudes in this $k \times k$ window and concatenate them into a vector of dimension $n = 2k^2$ for each feature. This is the dataset on which we perform PCA. Observe that even for a 30×30

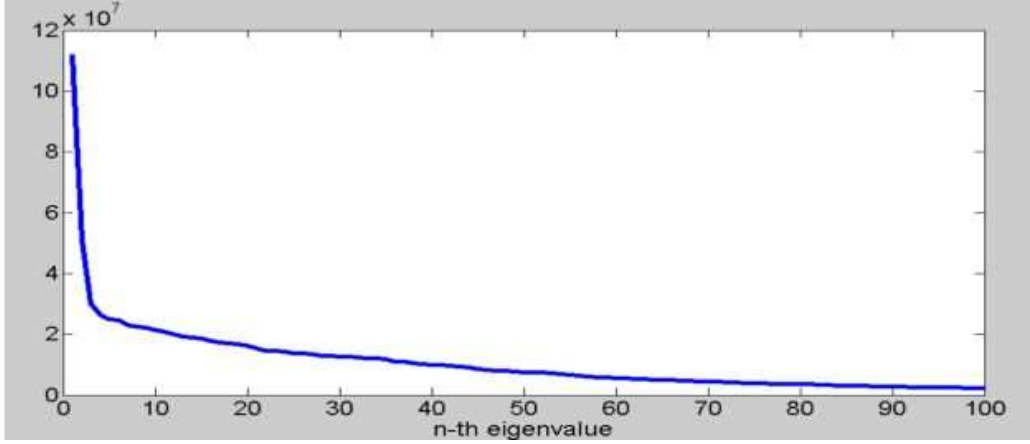


Figure 11: Plot of 100 largest eigenvalues using 31×31 (1922 dimensional training features) window for PCA algorithm.

window, the associated covariance matrix for which we need to do eigenvalue decomposition has dimensions 1800×1800 . Since our only concern is some subset of the largest eigenvalues, we can use numerical techniques that avoid the full eigendecomposition. Our current implementation uses the Matlab version of the Implicitly Restarted Arnoldi Method. We tested this method on an Eros dataset. In Fig. 11 we show a plot the 100 largest eigenvalues using a 31×31 window (i.e. $n = 1922$ in the notation above) Observe that the rapid decrease gives some evidence that as few as 30 principal components are necessary.

After the projection matrix is computed from the training set, subsequent computation of features descriptors is straightforward. We construct for each feature the $n = 2k^2$ original descriptor as described and use the matrix P computed during training to project it into a much smaller subspace. In Fig. 12 we show matching results using the first 10, 30 and 50 principal components as well as the result of the base algorithm without PCA. Note that for the 30 and 50 dimensional cases, performance is nearly identical to the non-PCA algorithm, and actually show more valid matches. The 10 dimensional case has many more outliers, but even here, the majority of matches are correct, and most outliers will be removed after applying the geometric constraints described in Sect. 2.3.3.

PCA has a number of advantages. It is a sparse representation of the most relevant data components. This means not only less storage and a smaller FCAT but potentially less sensitivity to image noise. Furthermore, since the data is naturally organized in terms of relevance (i.e. the n th element of the feature descriptor corresponds to data projection onto the eigenvector of n th largest eigenvalue), it follows that matching can be done much more efficiently. Instead of a simple Euclidean match, we can use a lexicographic approach that rejects incorrect matches after comparison of just the first few elements of the descriptor. However, performance will depend heavily on the training set and how representative it is for subsequent imagery to be used for the LMT. If there is significant bias in the training phase, performance may be poor.

We are also exploring the use of Independent Component Analysis (ICA), a technique in which

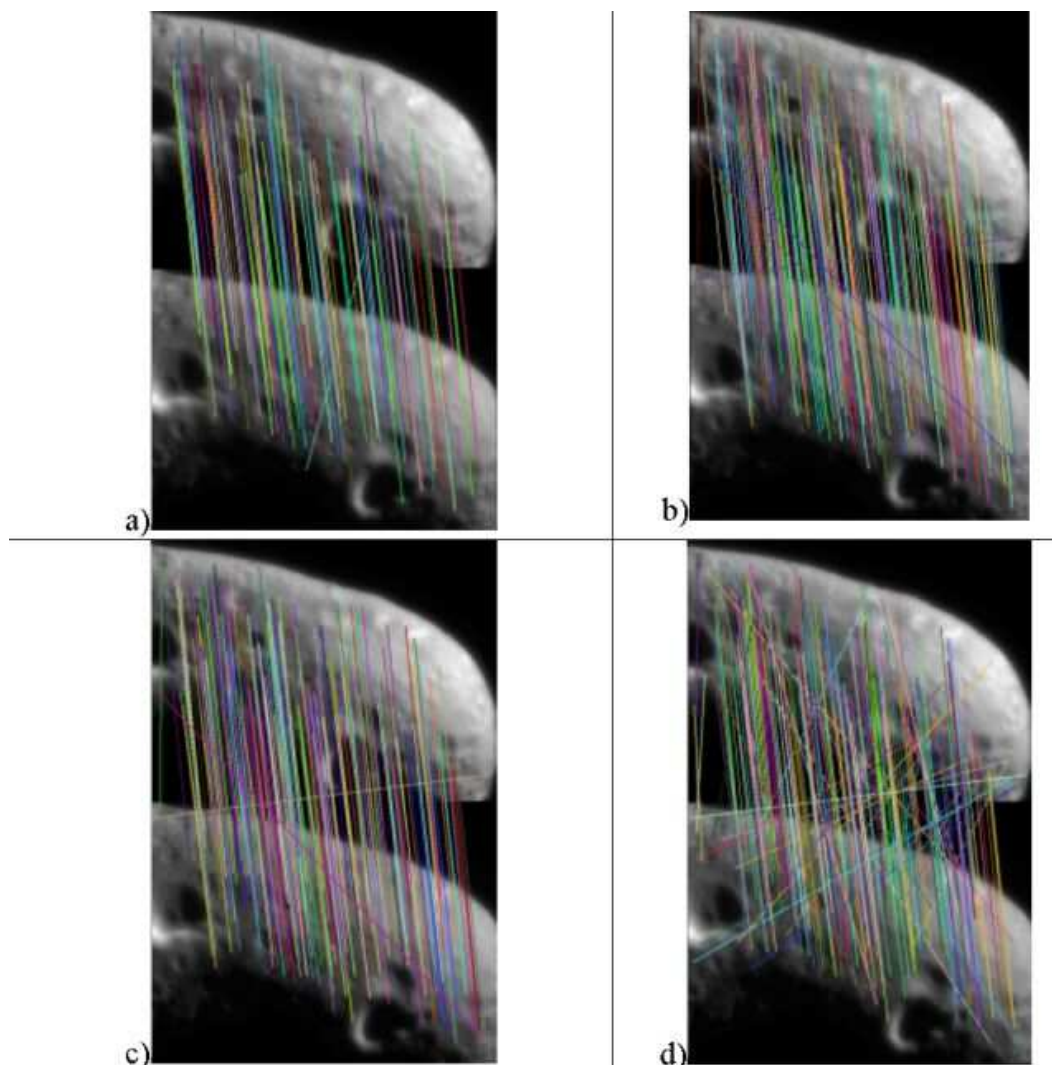


Figure 12: Matching results using a) the standard non-PCA algorithm, b) PCA with 50 dimensions, c) PCA with 30 dimensions, d) PCA with 10 dimensions.

the basis decomposition of the original vector space is not orthonormal, but aligned more naturally to the most relevant directions of information content. Our evaluation of this technique is still in the earliest stages.

3 Additional Vision Based Products

We describe in brief, some of the additional vision capabilities that can be applied to small body navigation. We start with the assumption that feature points are matched between overlapping frames or that landmarks have been identified in the catalog.

We have already described some of the 3D reconstruction techniques available to us. Generally speaking, scene reconstruction from two or more images with unknown camera motion is referred to in computer vision as the Structure from Motion (SfM) problem. There are many variants to the E matrix technique already described, including multi-frame techniques using a tensor constraint on image coordinates of matched points. In general, more robust techniques incorporate a non-linear optimization over the parameters of the relative camera motion, and possibly over the scene structure. We cite one survey paper [11] covering some classical techniques.

If the 3D coordinates of data points are also known in some fixed frame, localization of the camera in that frame can be accomplished very accurately. In our case, the 3D information is contained in FCAT and the image information in LMT. Given n 3D-2D point correspondences, the problem of localizing the camera in the world coordinate frame is known in computer vision as the N-Point Pose problem. There is an extensive body of literature on this subject as well. We refer the reader to [12] for an overview and numerous references. In Fig. 13, we show the localization error for position estimation using synthetic data of 10 points acquired by a 90 degree field of view camera 500m above a surface. We assume $\sigma = 0.5$ pixel error in image feature localization and error in map knowledge varying from $\sigma = 0$ to $\sigma = 10$ meters. Recovered position error for the camera is reported as absolute error in the body frame. If q_0 is the true attitude and q_r is the recovered attitude, then the angle associated with $q_{error} = q_r^{-1}q_0$ is reported as the attitude error. We ran this simulation over 100 trials using the noise levels indicated and report the RMS errors in Fig. 13. The absolute position estimate (using FCAT and LMT) and relative motion estimate (using PFT) can be used as a sanity check for the state estimator.

4 Conclusion and Future Work

Our adaptation of David Lowe’s SIFT work has shown some initial promise. We have produced good matching results in relatively easy cases. Our attempts at enhancing the algorithm to handle greater variation in viewpoint and lighting have met with some initial success, but more work is required. At present, we are focusing on PCA and ICA methods and anticipate that these may have better invariance properties in the small body setting. While the SIFT-like signature approach is worth further study, we also intend to explore other avenues. These include an extension of the SIFT framework to 3D structures. Since we have SfM techniques in place, we may be able to examine local 3D structure directly and develop feature descriptors based on both intensity

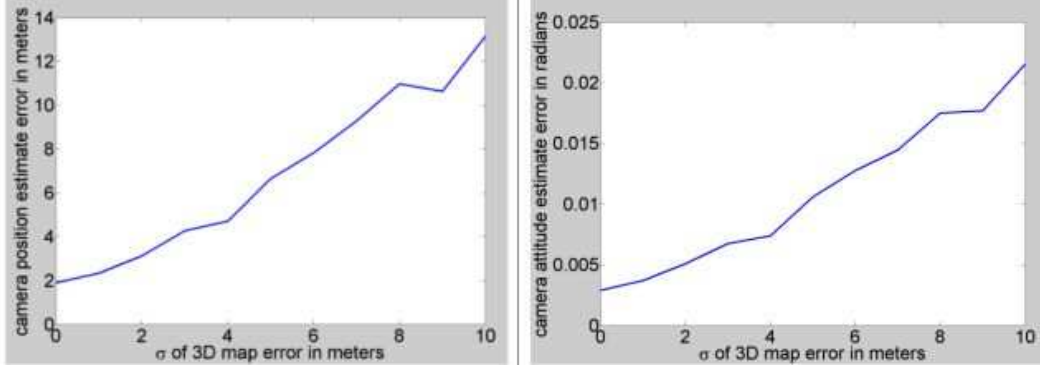


Figure 13: Position and attitude error vs. errors in 3D map knowledge of features for simulated camera 500 meters from surface. s is the standard deviation of Gaussian noise added to each coordinate.

information and structure. We also intend to explore the frequency domain more directly than the scale space formalism accomplishes to see if signatures can be computed directly in this arena. Another potentially productive area is the use of image transformation other than bandpass which capture local scene structure. We have started experimenting with scaled entropy images as a first step.

5 Acknowledgments

This research was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration and funded through the internal Research and Technology Development program.

References

- [1] Y. Cheng, A. E. Johnson, L. H. Matthies, and C. F. Olson, "Optical landmark detection for spacecraft navigation," in *Proceedings of the 13th Annual AAS/AIAA Space Flight Mechanics Meeting*, (Ponce, Puerto Rico), February 2003.
- [2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [3] C. Harris and M. Stephens, "A combined corner and edge detector," in *Fourth Alvey Vision Conference*, (Manchester, UK), pp. 147–151, 1988.
- [4] T. Lindeberg, "Scale-space theory: A basic tool for analyzing structures at different scales," *Journal of Applied Statistics*, vol. 21, no. 2, pp. 224–270, 1994.
- [5] J. J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, pp. 363–396, 1984.

- [6] K. Mikolajczyk and C. Schmid, “An affine invariant interest point detected,” in *European Conferenc on Computer Vision*, (Copenhagen, Denmark), pp. 128–142, 2004.
- [7] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceedings of the IEEE Conf. on Computer Vision*, (Bombay, India), 1998.
- [8] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
- [9] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.
- [10] Y.Ke and R. Sukthankar, “Pca-sift: A more distinctive representation for local image descriptors,” in *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, (Washington, DC), 2004.
- [11] T. Huang and A. Netravali, “Motion and structure from feature correspondences: A review,” *Proceedings of the IEEE*, vol. 82, no. 2, 1994.
- [12] C.-P. Lu, G. Hager, and E. Mjolsness, “Fast and globally convergent pose estimation from video images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 610–622, 2000.